
Gestion des mises à jour concurrentes dans des jeux de données géographiques répartis

Christelle Pierkot^{1,2,3}, Sébastien Mustière¹

¹IGN – COGIT
2-4 Avenue Pasteur
94160 Saint Mandé

²Univ. Paul Sabatier - IRIT
118, Route de Narbonne
31062 Toulouse cedex 9

³EADS - DS
6, voie l'Occitane
31676 Labège

{christelle.pierkot, sebastien.mustiere}@ign.fr

RÉSUMÉ : La mise à jour concurrente des jeux de données géographiques pose des problèmes de cohérence lors de l'intégration des évolutions provenant de sources multiples. Cela est d'autant plus délicat dans un environnement de réplication optimiste des données avec une mise à jour effectuée en parallèle, sans verrouillage de l'écriture et parfois en mode déconnecté.

Dans ce contexte, notre objectif est d'assurer la cohérence des jeux de données à différents niveaux selon la finalité de la mise à jour telle qu'une prise de décision rapide, ou la production de données. Nous proposons une stratégie d'intégration cohérente des mises à jour basée sur un contrôle de concurrence couplé à des sessions de mises à jour.

Le contrôle de concurrence sert à détecter les mises à jour conflictuelles et à les réconcilier si nécessaire. L'objectif du protocole de réconciliation est de proposer la « meilleure solution » lorsqu'un conflit est détecté. Le résultat, dépendant du niveau de cohérence souhaité, est choisi grâce à un ensemble de métadonnées qui est fourni avec les données et les évolutions.

Les sessions de mise à jour sont quant à elles utilisées pour limiter les actions sur les données de l'utilisateur et pour avoir des états du jeu de données stables à des instants particuliers.

ABSTRACT : Updating concurrent spatial dataset gives some consistency problems. This is trickier in a lazy replicated environment where the update is made in parallel, without locking mechanism and sometimes in a disconnected mode.

In this context, our main objective is to maintain the consistency of spatial datasets by taking into account the finality of the update (production decision making...). We propose an integration strategy based on a concurrency control coupled with integration sessions of evolutions.

Concurrency control is dedicated to check the conflicting updates and to reconcile them if necessary. The result depended on the level of consistency that the final user need, is done with the help of metadata which are joined with the data and updates.

Updating sessions are used to limit actions on the user dataset and to have stables dataset at particular times.

MOTS-CLES : Mise à jour, concurrence, cohérence, réplication optimiste

KEYWORDS : Update, concurrency, consistency, lazy replication

1. Contexte et Introduction

De nos jours, les nouvelles technologies permettent le développement de réseaux de communication grâce auxquels les utilisateurs peuvent échanger facilement de l'information. Un contexte applicatif est celui d'une mission militaire où des unités fixes et mobiles sont réparties sur différents sites et doivent coopérer. Pour faciliter la collaboration, un réseau de communication est mis en place entre les différents acteurs afin qu'ils puissent partager leurs données. L'information n'est pas centralisée et les utilisateurs peuvent ponctuellement être déconnectés du réseau, par exemple lorsqu'une unité mobile va faire des mesures sur le terrain.

Les acteurs d'une mission militaire utilisent des données spatiales vectorielles dérivées d'une base de référence et répliquées sur les différents sites afin qu'elles soient localement toujours disponibles. La mise à jour de ces données est effectuée en parallèle et sans verrous, en fonction des rôles et besoins de chacun et par conséquent avec des niveaux de détail et des qualités différentes. Pour assurer une collaboration efficace, des échanges multidirectionnels des informations de mise à jour sont régulièrement effectués entre les différents acteurs. Des difficultés se produisent lorsqu'un utilisateur veut intégrer ces nombreuses évolutions dans son jeu de données personnel.

Pour reformuler ce contexte avec les termes consacrés en base de données, notre recherche se situe dans le cadre d'une application collaborative asynchrone utilisant un protocole de réplication optimisée de données géographiques dans un contexte de prise de décision. Plus particulièrement, nous nous intéressons à la mise à jour d'un jeu de données géographique vectoriel par des évolutions provenant de sources multiples. Ces mises à jour ne sont pas nécessairement pertinentes pour l'utilisateur final mais surtout, elles peuvent être en conflit et par conséquent créer des incohérences lors de l'intégration.

Pour résoudre ces différents problèmes, nous proposons une stratégie d'intégration des mises à jour basée sur un contrôle de concurrence couplé à des sessions de mises à jour. Les éventuels conflits sont détectés et un protocole de réconciliation basé sur des métadonnées adéquates est alors exécuté pour résoudre les problèmes de cohérence.

Dans cet article nous expliquons d'abord pourquoi l'utilisation d'une infrastructure de données spatiale est profitable à l'intégration des mises à jour. Ensuite, nous exposons les problèmes de cohérence qui peuvent survenir lors de la mise à jour de données géographiques répliquées. Puis, nous décrivons la stratégie d'intégration des mises à jour que nous avons mise en place. En particulier, nous détaillons le protocole de vérification de la concurrence et le processus de réconciliation. Nous discutons également de la nécessité d'effectuer des sessions de mise à jour plus ou moins fréquentes en fonction de la cohérence qui est souhaitée.

2. Infrastructure de données spatiales et Métadonnées

Une infrastructure de données spatiale (SDI) est une initiative prévue pour créer un environnement dans lequel tous les acteurs peuvent coopérer les uns avec les autres et interagir, pour mieux atteindre leurs objectifs à différents niveaux politiques et administratifs (Rajabifard et al. 2001). Une SDI permet donc de partager plus efficacement l'information entre plusieurs d'acteurs participant à un objectif collectif comme une mission militaire dans notre contexte. En particulier, on peut y définir un schéma commun pour le transfert des évolutions permettant ainsi une meilleure gestion des mises à jour quel que soit l'endroit où elles doivent être intégrées dans l'infrastructure. De ce fait, des règles peuvent être employées pour définir comment les utilisateurs doivent intégrer leurs évolutions et les mises à jour provenant d'autres acteurs.

Par ailleurs, les métadonnées aident les utilisateurs de données spatiales à trouver les données dont ils ont besoin et à déterminer comment les employer au mieux (Nebert 2004). Il existe un lien étroit entre les acteurs, les données et les évolutions répartis entre plusieurs sites et l'utilisation des métadonnées est une solution pour gérer cela efficacement (Pierkot et al. 2005). Nous pouvons donc supposer que les métadonnées peuvent aider les personnes qui échangent des évolutions spatiales à identifier les mises à jour pertinentes et à déterminer comment les intégrer au mieux.

Plusieurs normes de métadonnées consacrées aux données spatiales ont été définies pour assurer l'interopérabilité entre des utilisateurs manipulant de l'information géographique (CEN 1998 ; FGDC 2000 ; ISO 2003). Celle qui retient le plus l'attention de nos jours est l'ISO 19115 (ISO 2003). Cette norme définit un schéma pour décrire l'information géographique et les services associés. Elle fournit beaucoup d'informations telles que l'identification, la qualité ou la distribution des jeux de données spatiaux mais elle est difficile à exploiter du fait du très grand nombre d'éléments à gérer. Néanmoins, il est possible de créer des profils par extension et restriction de la norme. De plus en plus d'organismes exploitent cela pour créer leur profil communautaire, comme l'armée française avec METAFOR (METAFOR 2005).

Les normes de métadonnées décrivent précisément les données mais peu leur mise à jour. Nous avons ainsi défini un profil de métadonnées dénommé MUMSDI (Metadata for Updating Military Spatial Data Infrastructure) pour faciliter le partage des évolutions résultant des mises à jour de divers jeux de données vectoriels dans un contexte militaire. MUMSDI restreint d'une part l'ISO 19115 en enlevant toutes les informations inutiles pour les données d'évolutions et en précisant un domaine plus restrictif sur des éléments existants. D'autre part, MUMSDI étend la norme en ajoutant plus de renseignements sur la qualité des évolutions, en particulier pour déterminer l'adéquation aux besoins (voir (Pierkot et al. 2006) pour plus de détails).

Dans le cadre de ce travail, nous nous sommes donc appuyés sur un SDI dans lequel un réseau de communication est défini, les utilisateurs avec leurs différents rôles sont connus, les métadonnées utiles sont spécifiées et où l'échange des mises à

jour peut être mené grâce à une stratégie commune. La figure 1 montre un exemple des échanges et des rôles que l'on trouve dans cette infrastructure militaire. L'infrastructure y est composée de deux sites dans lesquels les acteurs possèdent des rôles différents (producteurs légers, lourds ou simples utilisateurs). Les tâches à accomplir diffèrent en fonction du rôle : la fonction principale des producteurs lourds est de préparer les futurs jeux de données alors que celle des producteurs légers est de fournir les mises à jour aux utilisateurs qui doivent prendre des décisions rapides et être réactifs. Les mises à jour saisies en mode déconnecté sont appliquées localement immédiatement mais envoyées ultérieurement aux autres acteurs lors de la reconnexion au réseau.

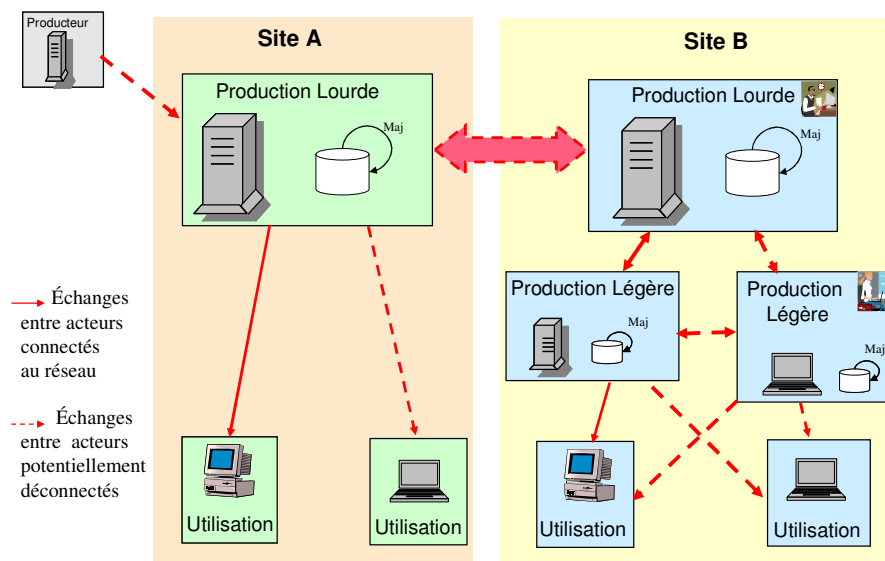


Figure 1. Rôles et échanges des acteurs dans notre infrastructure militaire

3. Réplication, concurrence et problèmes de cohérence

La réplication optimiste est un domaine de recherche très actif en bases de données et en systèmes (Cederqvist et al. 2001; Kermarrec et al. 2001, Oracle 2003; Seshadri et al. 2000). Les principales caractéristiques d'un système de réplication optimiste sont les suivantes :

- Chaque site de réplication possède une copie des objets partagés qu'il peut modifier librement et à tout moment.
- La mise à jour est exécutée localement immédiatement puis envoyée aux autres sites pour réexécution ultérieure.
- Toutes les écritures sont acceptées a priori, ce qui implique que les mises à jour sont potentiellement concurrentes et sources d'incohérences. Le terme incohérence est ici pris dans le sens « divergence des copies » créant de fait un état instable du système de réplication.

- La résolution est effectuée a posteriori lors de la synchronisation qui détecte les éventuels conflits. La réconciliation des écritures divergentes est alors exécutée pour obtenir la convergence du système. En réplication optimiste, on dit qu'un système converge lorsque toutes les opérations ont été propagées sur tous les sites et que les répliques sont identiques. On dit alors que le système de réplication est dans un état cohérent.

En information géographique, le terme cohérence n'a pas le même sens qu'en réplication optimiste. En effet, le terme cohérence ne signifie pas ici convergence stricte mais doit plutôt être pris dans le sens « s'assurer que les données produites ne représentent pas des visions incompatibles du monde réel ». Trois grands types de cohérence en information géographique existent :

- La cohérence des modèles qui est relative aux choix de modélisation. Elle concerne par exemple le mode géométrique utilisé (raster, vecteur).
- La cohérence des schémas qui est relative aux choix de représentation de l'information. Elle concerne par exemple les différences de granularité des attributs entre deux classes, ou le fait qu'une information peut être représentée comme une classe ou comme un attribut.
- La cohérence des données qui est relative aux instances elles mêmes. On en distingue deux catégories : la cohérence spatiale qui concerne la géométrie et la topologie des objets géographiques et la cohérence thématique qui concerne les attributs non spatiaux. Un exemple de cohérence souhaité est qu'une route ne doit pas traverser un bâtiment.

Dans le cadre de notre étude, la cohérence ne sous entend donc pas convergence des copies (dans le sens égalité) comme en réplication optimiste mais plutôt compatibilité des copies telle qu'elle est définie en information géographique. Par ailleurs, nous restreignons notre problème à l'étude de la cohérence des données, et non des modèles ou schémas qui est supposée traitée dans l'infrastructure.

Différents niveaux de cohérence peuvent être souhaités en fonction des rôles et objectifs des acteurs dans l'infrastructure. Par exemple dans notre contexte, l'objectif premier en production légère est de pouvoir rapidement prendre des décisions et d'être très réactif. La cohérence souhaitée sera donc ici plutôt faible car on privilégie la quantité à la qualité et cohérence des données. En revanche, en production lourde, l'objectif est de préparer les futurs jeux de données. La cohérence doit donc être forte car on privilégie la qualité plutôt que la quantité d'information. Enfin, un niveau de cohérence intermédiaire entre production lourde et production légère est souhaité car le but ici est de remonter l'information saisie sur le terrain au niveau du producteur lourd.

Notre travail s'attache donc à la gestion de la cohérence, à différents niveaux, des répliques d'un jeu de données géographiques, lors de l'intégration d'évolutions provenant de sources distantes. Plusieurs problématiques en découlent directement. Premièrement, il faut définir ce qu'est une incohérence et déterminer comment les détecter. Secondement, il faut élaborer des mécanismes de réconciliation des mises à

jour conflictuelles, c'est-à-dire déterminer quelle est la meilleure solution en fonction du niveau de cohérence souhaité, en fonction de la qualité des mises à jour mais aussi en tenant compte des besoins des utilisateurs. Enfin, il faut prévoir l'intégration des mises à jour dans le jeu de données utilisateur.

4. Stratégie d'intégration des mises à jour

Lorsqu'un jeu de données est mis à jour grâce à plusieurs ensembles d'évolutions provenant de sources distinctes, il faut gérer la concurrence entre les mises à jour, et entre les données et les mises à jour, afin de limiter les incohérences. Le problème de la concurrence des données répliquées a été étudié dans les communautés systèmes (Cederqvist et al. 2001 ; Kermarrec et al. 2001) et bases de données (Oracle 2003 ; Seshadri et al. 2000). Des protocoles ont été définis, mais la plupart d'entre eux supposent l'existence d'un serveur de référence centralisant les données. Les mécanismes employés dans l'information géographique pour gérer la concurrence sont l'approche « check in – check out » (ESRI 2004) et le versionnement (Intergraph 2003). Dans le premier cas, les données sont verrouillées aux autres utilisateurs jusqu'à ce que les mises à jour aient été intégrées, et le second nécessite d'utiliser un serveur centralisé contenant les données de référence. Aucune de ces méthodes n'est donc complètement appropriée à notre contexte distribué où les données sont répliquées sur chaque site, peuvent évoluer en parallèle et différemment, et doivent être à la disposition des utilisateurs.

Nous proposons donc une stratégie d'intégration des mises à jour dont l'objectif est d'aider l'intégration d'un flot continu d'évolutions dans un jeu de données particulier, sans rechercher une convergence stricte. Notre méthode est basée sur un contrôle de concurrence couplé à des sessions de mises à jour (figure 2). Durant le contrôle de concurrence, les évolutions multi-sources sont comparées entre elles et avec les données de l'utilisateur afin de détecter les éventuels conflits et d'effectuer une réconciliation si besoin est. Les évolutions non conflictuelles sont ensuite intégrées dans le jeu de données utilisateur à chaque session de mise à jour.

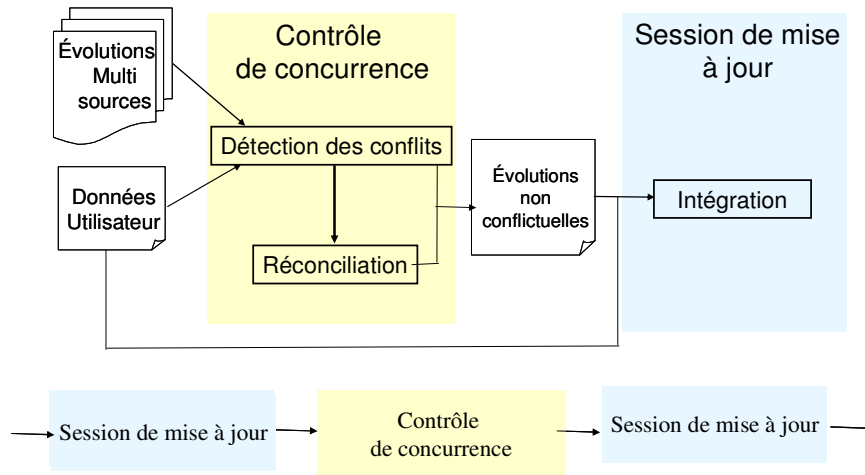


Figure 2. *Stratégie d'intégration des mises à jour*

4.1. Sessions de mise à jour

L'existence de sessions de mise à jour par rapport à une intégration en continu des mises à jour apporte deux bénéfices majeurs. Premièrement, elles permettent de limiter les actions sur les données de l'utilisateur. En effet, les évolutions multi-sources arrivant en continu, il est possible de recevoir successivement plusieurs mises à jour concernant le même objet. Si on intégrait immédiatement la mise à jour dans le jeu de données utilisateur, il faudrait parfois défaire et refaire les actions quand une nouvelle mise à jour se présenterait, avec tous les problèmes de recalage que cela implique. Ensuite, elles nous permettent d'avoir des états du jeu de données considérés comme stables à des instants donnés, états dans lesquels on peut facilement revenir en cas de besoin si les journaux des modifications sont conservés.

La fréquence des sessions de mise à jour est étroitement liée à la cohérence (Figure 2, partie 2). En effet, les utilisateurs et producteurs légers ont besoin d'obtenir rapidement de l'information quelle que soit sa qualité. Il est donc souhaitable que la durée entre les sessions de mises à jour soit courte afin que les jeux de données soient mis à jour le plus vite possible quitte à défaire et refaire souvent les opérations, ce qui est un moindre mal étant donné que la cohérence souhaitée est faible. En revanche, au niveau du producteur lourd, les informations doivent être cohérentes et de qualité, et des actions interactives peuvent être réalisées pour cela. Il est donc préférable que le temps entre les sessions de mise à jour soit plus élevé afin de minimiser les opérations sur les données.

Entre chaque session de mise à jour, les évolutions provenant des sources multiples arrivent en flux continu et peuvent créer des incohérences si elles sont intégrées sans attention particulière. En effet, les évolutions provenant des sources multiples peuvent soit concerner un objet qui est déjà présent dans le jeu de données d'utilisateur (intégré lors d'une session précédente par exemple), soit avoir été

envoyée par plusieurs producteurs (avec des qualités différentes par exemple). Ainsi, entre les sessions de mise à jour, les évolutions doivent être traitées pour détecter les éventuels conflits à résoudre, qu'ils soient entre les mises à jour, ou entre les mises à jour et les données. Le module de contrôle de concurrence permet de détecter ces conflits et de proposer des mécanismes de réconciliation si besoin est.

4.2. Contrôle de concurrence

La figure 3 montre le processus permettant de gérer la concurrence entre un ensemble de données utilisateur et un flot continu d'évolutions. Cette méthode appelée contrôle de concurrence est divisée en deux phases distinctes mais fortement liées : vérification de la concurrence et réconciliation.

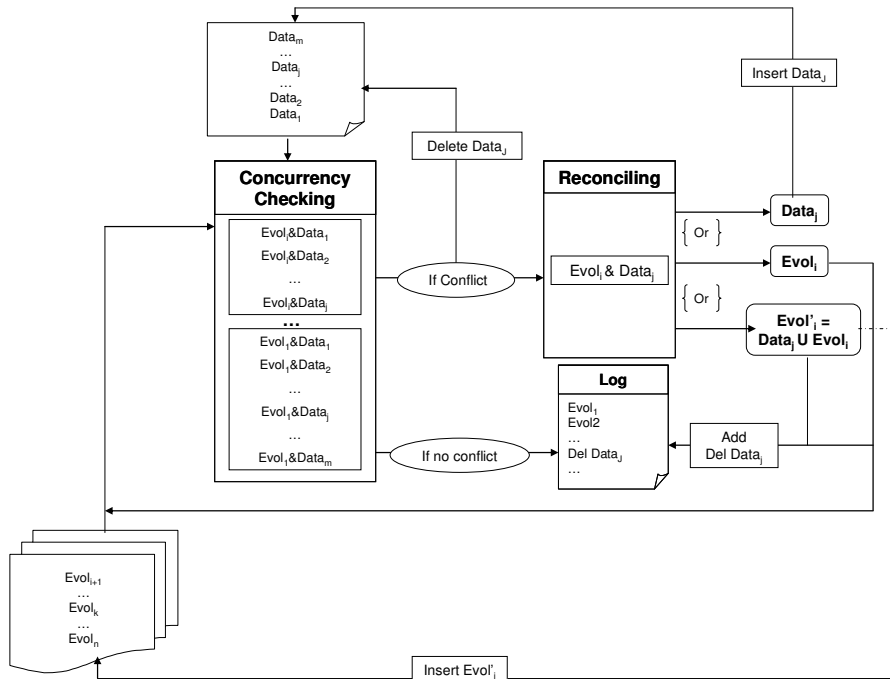


Figure 3. Protocole de contrôle de concurrence

Dans le cadre de notre étude, un conflit est provoqué par une mise à jour qui viole la cohérence des données tel que cela est défini en information géographique. Il peut être déclenché soit par une mise à jour impliquant deux objets du monde réel et causé par des niveaux de détails différents ou des erreurs et imprécisions de saisies (par exemple, la création d'une route sur une maison), soit par deux mises à jour relatives au même objet du monde réel mais issues de sources distinctes. Ces conflits sont causés par des différences dans la géométrie ou dans les valeurs

d'attribut ou encore par la manière dont les évolutions sont décrites. Typiquement, un utilisateur peut avoir fait une modification et un autre utilisateur, une suppression suivie d'une création pour décrire la même évolution.

4.2.2. Détection des conflits

Durant la phase de détection, trois types de conflits peuvent être repérés.

Pour identifier les conflits dus à des modifications ou des suppressions concurrentes, nous nous appuyons sur les identifiants fournis avec les évolutions et les données. En effet, dans notre contexte les données possèdent initialement des identifiants uniques et pérennes, générés à partir de la géométrie et des attributs des objets spatiaux. Les mises à jour de type modification et suppression ayant été saisies à partir de données existantes, possèdent une référence à ces identifiants que l'on peut comparer pour détecter les conflits.

Les conflits de recouvrement entre objets différents sont quant à eux détectés grâce à des tests d'intersection entre les géométries. Dans ce cas, les objets concernés par la mise à jour ne sont pas les mêmes et possèdent par conséquent des identifiants différents que nous ne pouvons pas utiliser pour vérifier la concurrence.

Enfin, les conflits dus à la création d'un objet sont détectés grâce à des techniques d'appariement géométrique. Les identifiants ne sont pas non plus exploitables ici car ils sont générés lors de l'intégration, et n'existent donc pas encore lors de la détection de conflits.

Si aucun conflit n'est décelé, l'évolution est stockée dans un journal qui contient les actions à exécuter lors de la prochaine session de mise à jour. Le processus analyse alors l'évolution suivante. En revanche, lorsqu'un conflit est détecté la vérification de la concurrence s'arrête, la donnée de l'utilisateur en conflit est supprimée du jeu de données et le processus de réconciliation est lancé.

4.2.3. Réconciliation

Le but du processus de réconciliation est de proposer l'option la plus appropriée entre les données en conflit. La réconciliation dépend du niveau de cohérence souhaité et de l'équilibre voulu entre qualité et quantité. Par exemple, au niveau des producteurs légers et des utilisateurs, la cohérence est faible et dépend principalement des besoins des utilisateurs. Il n'est donc pas possible de faire converger les répliques. En revanche, la cohérence entre producteurs lourds est forte et il est souhaitable qu'il y ait convergence stricte des répliques.

Le protocole de réconciliation que nous avons mis en place utilise des métadonnées et s'appuie sur le modèle que nous avons défini dans (Pierkot et al. 2005) pour établir les relations entre les acteurs, données et évolutions. Trois types de métadonnées y sont disponibles (cf. figure 4) :

- Les métadonnées fournies avec les jeux de données. Elles sont conformes à la norme ISO 19115 et fournissent des informations notamment sur l'identification et la qualité des ressources.
- Les métadonnées fournies avec les évolutions. Elles sont conformes au profil de métadonnées MUMSDI que nous avons créé pour la gestion des évolutions dans un contexte d'infrastructure militaire. Ces métadonnées nous donnent de l'information sur les mises à jour partagées dans une infrastructure militaire à des granularités différentes (évolution élémentaire, ensemble d'évolution...), ainsi que des renseignements concernant la qualité et la fiabilité des mises à jour.
- Les métadonnées des utilisateurs se distinguent en deux catégories selon que la convergence est souhaitée ou non : les besoins et les contraintes de cohérence des utilisateurs. Les contraintes de cohérence sont appliquées au niveau des producteurs lourds. Elles sont définies dès le début de la mission et ne peuvent changer. En revanche, les besoins des utilisateurs dépendent du site, du travail à effectuer et de l'urgence de la situation (période de crise ou campagne de relevé) et peuvent évoluer au cours de la mission.

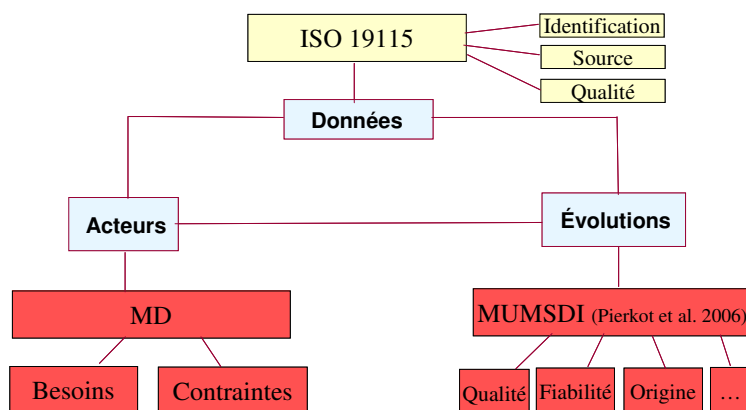


Figure 4. *Métadonnées utilisées lors de la réconciliation*

Les métadonnées fournies avec les évolutions et les données aident le processus à effectuer le meilleur choix, en analysant des champs comme la qualité ou l'origine des mises à jour. Ainsi, le processus peut choisir entre l'évolution et la donnée de l'utilisateur celle qui est la meilleure, ou peut créer une nouvelle évolution combinée. Une nouvelle évolution est créée quand une partie de l'évolution et une partie de la donnée donnent une meilleure solution qu'une ou l'autre prise individuellement. Par exemple, supposons que la donnée utilisateur ait une géométrie plus précise et que l'évolution ait des attributs plus fiables. Si la contrainte est d'avoir les données les plus précises alors une nouvelle mise à jour est créée avec la géométrie de la donnée et les attributs de l'évolution. Finalement, en sortie du processus de réconciliation, on obtient la donnée initiale ou l'évolution ou une nouvelle mise à jour :

- Si la donnée initiale est choisie ($Data_j$ dans la figure 3), alors elle est réinjectée dans le jeu de données, et la méthode redémarre avec une nouvelle évolution.
- Si l'évolution est choisie ($Evol_i$ dans la figure 3) alors la méthode reprend la vérification de la concurrence entre l'évolution et le reste des données afin de s'assurer qu'il n'existe plus de conflits. L'action première « effacer la donnée » est également stockée dans le journal afin qu'elle soit jouée lors de la prochaine session de mise à jour.
- Si une nouvelle mise à jour est créée ($Evol_i$ dans la figure 3) alors elle est mise à la fin de la liste des évolutions courantes et la méthode est relancée avec une nouvelle évolution. L'action première « effacer la donnée » est également stockée dans le journal pour la prochaine session de mise à jour.

La méthode continue tant qu'il y a des évolutions à examiner, ou jusqu'à la prochaine session de mise à jour. Les mises à jour saisies par l'utilisateur lui-même entre deux sessions sont considérées comme les autres évolutions et doivent être examinées avant toute intégration.

A la sortie de ce processus, les évolutions stockées dans le journal sont assurées de ne pas être en conflit avec les données de l'utilisateur mais peuvent être concurrentes les unes avec les autres. En effet, du fait de la provenance multiple des mises à jour, il est possible qu'une évolution parvienne plusieurs fois à l'utilisateur. Nous appliquons donc également le protocole de contrôle de concurrence entre toutes les évolutions afin de contrôler la concurrence entre les mises à jour et proposer des solutions de réconciliation si nécessaire. Finalement, avant une session de mise à jour, les évolutions présentes dans le journal des modifications ne posent plus de problèmes de cohérence et peuvent donc être intégrées.

5. Conclusion

Dans cet article nous avons proposé une méthode pour gérer un flux continu de mises à jour provenant de sources multiples à intégrer dans un jeu de données utilisateur. Nous avons d'abord montré la nécessité de définir une infrastructure de données spatiale dans laquelle des règles peuvent être appliquées, puis nous avons spécifié les métadonnées requises pour la gestion des évolutions. Enfin, nous avons défini une stratégie globale pour la gestion de la concurrence entre les évolutions et les données d'un utilisateur particulier et nous avons proposé des solutions de réconciliation en cas de conflit. Des tests ont été réalisés avec des données militaires réelles et des mises à jour simulées qui nous encouragent à continuer dans cette voie.

Il est toujours préférable d'automatiser un maximum le protocole de réconciliation notamment lorsque la convergence est souhaitée. Dans le meilleur des cas, il est souhaitable que le processus de réconciliation soit automatique afin d'éviter différents choix faits par des utilisateurs distincts qui ont les mêmes besoins mais qui n'ont pas la même perception du monde. Mais parfois, les métadonnées fournies ne sont pas suffisantes pour définir le meilleur choix en fonction des exigences d'utilisateur, et un processus interactif est alors nécessaire pour aider

l'utilisateur à prendre une décision. Trois approches ont été identifiées le processus de réconciliation : 1) automatique quand il y a suffisamment d'information pour s'assurer qu'un choix est meilleur que les autres, 2) interactif mais guidé quand plusieurs choix peuvent être acceptables ou, 3) totalement laissé à la charge de l'utilisateur quand aucune solution satisfaisante ne peut être trouvée. La difficulté principale dans l'automatisation est de définir quand le résultat est considéré comme convenable, ce qui fera l'objet des travaux futurs pour améliorer le processus de réconciliation en assurant le meilleur compromis entre automatisation et interaction.

6. Bibliographie

- CEN, 1998, Geographic Information European Prestandards, Euro-norme Voluntaire for Geographic Information –Data description- Metadata. ENV 12657, European Committee for Standardization – CEN/TC287.
- Cederqvist P. et Pesch R. 2001. Version Management with CVS. www.cvshome.org/docs/manual
- ESRI 2004. Versioning. An ESRI ® Technical Paper. January 2004. www.esri.com
- FGDC, 2000, Content Standard for Digital Geospatial Metadata, version 2.0. Document FGDC-SDT-001-1998, Federal Geographic Data Committee, Metadata Ad Hoc Working Group. <http://www.fgdc.gov/>
- Kermarrec, A., Rowstron, A., Shapiro, M., and Druschel, P. 2001. The icecube approach to the reconciliation of divergent replicas. *In Proceedings of the 20th ACM Symposium on Principles of Distributed Computing (PODC 2001)*, Newport, Rhode Island (USA).
- Intergraph 2003. White paper on Versioning, Lineage, Timestamps and Temporal Databases. Intergraph Corporation, 2003.
- ISO 2003. Geographic Information - Metadata. ISO 19115:2003, International Organisation for Standardisation.
- METAFOR, 2005, Gamme de produits CARGENE. Format de fichiers de métadonnées. République Française, Ministère de la défense. IGN/DT.TN/03.054
- Nebert D. 2004. Developing Spatial Data Infrastructures : *The SDI Cookbook, version 2.0*.
- Oracle 2003. Oracle® Database Advanced Replication 10g Release 1 (10.1) Part No. B10732-01 December 2003.
- Pierkot C, Mustiere S, Ruas A, Hameurlain H and Raynal L., 2005. Modelling Heterogeneous and Distributed Spatial Datasets in an Update Context. *Proceeding in the 22th International Cartographic Conference ICC 2005, Coruna, Espagne*.
- Pierkot C, Mustiere S, Ruas A, Hameurlain H, 2006. Using metadata to help the integration of several multi-sources set of updates. *Proceeding in the 9th Global Spatial Data Infrastructure Conference GSDI 2006, Santiago, Chile, GSDI Association Publications*.
- Rajabifard A. and Williamson I.P., 2001. Spatial Data Infrastructures: Concept, SDI Hierarchy and Future Directions. *Proc. of GEOMATICS'80 Conference, Téhéran, Iran*.
- Seshadri P., Garrett P. 2000. SQLServer for Windows CE - A Database Engine for Mobile and Embedded Platforms. *Int. Conf. on Data Engineering (ICDE), 2000*.